IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPEAL BRIEF – 37 C.F.R § 1.192

U.S. Patent Application 10/785,108 entitled:

"Dynamically Capturing Data Warehouse Population Activities for Analysis, Archival, and Mining"

Real Party in Interest: International Business Machines Corporation

Related Appeals and Interferences:

None

Status of Claims:

1. Claims 1-17 and 19-38 are pending.

2. Claims 1-17 and 19-38 are rejected.

3. Claims 1-17 and 19-38 are rejected under 35 U.S.C. 103(a) as being unpatentable over

   Thompson et al. (6,668,253) in view of Norcott (2003/0172091).

4. Claims 1-17 and 19-38 are hereby appealed.

Status of Amendments:

No After-Final Amendments were submitted after the Final Rejection of 03/09/2007.

Summary of Claimed Subject Matter:

(NOTE: All citations are made from corresponding Pre-Grant Publication 2005/0187991, which

contains the original specification, including the figures.)

Independent **claim 1** teaches a method for archiving task information obtained from a data-

warehousing environment comprising steps of: obtaining changes in operational metadata from

said data-warehousing environment (**see Abstract, Figure 1, and Paragraphs 20 and 26 of

Pre-Grant Publication 2005/0187991**), extracting task information from said operational

metadata (**see Abstract, Figure 1, and Paragraphs 20, 25 and 28 of Pre-Grant Publication**

2005/0187991), storing said extracted task information in a buffer, refreshing said buffer with changes in said operational metadata (**see Abstract, Figure 1, and Paragraphs 25, 26 and 28 of Pre-Grant Publication 2005/0187991**), and moving task information from said buffer to an archive, said archived task information used in data analysis and mining (**see Abstract, Figure 1, and Paragraphs 10, 12, 20, 25-27, 29, 30, 32, and 36 of Pre-Grant Publication 2005/0187991**).

Independent claim 10 teaches a method for capturing and recording task information obtained from a data-warehousing environment for analysis, archival, and mining comprising steps of: uniquely identifying each task within a run (**see Paragraph 22 of Pre-Grant Publication 2005/0187991**), selecting one or more of said uniquely identified tasks to monitor (**see Paragraphs 23-27 of Pre-Grant Publication 2005/0187991**), capturing data-warehousing population activities dynamically by obtaining operational metadata containing task information relevant to said selected task or tasks (**see Paragraphs 20, 21, 24, 27, 35 of Pre-Grant Publication 2005/0187991**), calculating changes in operational metadata, storing results of said calculating step in a buffer, and moving selected buffer data to an archive, said archive used in data analysis and mining (**see Abstract, Figure 1, and Paragraphs 10, 12, 20, 25-27, 29, 30, 32, and 36 of Pre-Grant Publication 2005/0187991**).

The present invention's independent claim 29 teaches a data-warehousing environment system for capturing and recording task information, said data-warehousing environment implemented in computer storage (**see Paragraphs 34, 35, and 37 of Pre-Grant Publication 2005/0187991**), said computer storage storing: task information extracted from operational

metadata (**see Abstract, Figure 1, and Paragraphs 25, 26 and 28 of Pre-Grant Publication 2005/0187991**), trigger mechanisms attached to said operational metadata (**see Paragraph 20 and Figure 1, element 102, 110, and 112**), staging table storing said task information (**see Paragraph 20 and Figure 1, element 116**), trigger mechanisms attached to said staging table (**see Paragraph 20 and Figure 1, element 114**), and an archive table storing task information from said staging table (**see Paragraph 20 and Figure 1, element 118**).

The present invention's independent claim 30 teaches an article of manufacture comprising a computer storage medium having computer readable program code embodied therein which implements the archiving of task information obtained from a data-warehousing environment comprising modules to execute the steps of (**see Paragraphs 34, 35, and 37 of Pre-Grant Publication 2005/0187991**): obtaining changes in operational metadata from said data-warehousing environment (**see Abstract, Figure 1, and Paragraphs 20 and 26 of Pre-Grant Publication 2005/0187991**), extracting task information from said operational metadata (**see Abstract, Figure 1, and Paragraphs 20, 25 and 28 of Pre-Grant Publication 2005/0187991**), storing said extracted task information in a buffer, refreshing said buffer with changes in said operational metadata (**see Abstract, Figure 1, and Paragraphs 25, 26 and 28 of Pre-Grant Publication 2005/0187991**), and moving task information from said buffer to an archive (**see Abstract, Figure 1, and Paragraphs 10, 12, 20, 25-27, 29, 30, 32, and 36 of Pre-Grant Publication 2005/0187991**).

Grounds of Rejection to be Reviewed on Appeal:

Claims 1-17 and 19-38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Thompson et al. (6,668,253) in view of Norcott (2003/0172091). <u>Claims 1-17 and 19-38 are hereby the subject of this appeal. Was a proper rejection made under 35 U.S. C. § 103(a) using existing USPTO guidelines?</u>

ARGUMENT:

<u>REJECTIONS UNDER 35 U.S.C. § 103(a)</u>

Claims 1-17 and 19-38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Thompson et al. (6,668,253) in view of Norcott (2003/0172091). To establish a prima facie case of obviousness under U.S.C. § 103, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest <u>all</u> the claim limitations. Additionally, the teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure (In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991)). Applicants contend, and as will be shown in the response below, that an improper rejection was given with regards to peding claims 1-17 and 19-38 using the Thompson et al. and Norcott references.

Thompson et al. teaches a system for enterprise information management comprising: a

5

data warehouse server; a transformation and staging server connected to the data warehouse

server for providing transformed and cleansed data to the data warehouse server; a data source

application connected to the transformation and staging server to provide data to the

transformation and staging server; a financial consolidation application connected to the

transformation and staging server for performing consolidation and reporting of financial data; a

web server connected to the data warehouse server; and a plurality of clients connectable to the

web server for accessing data from the data warehouse server via the web server.

With respect to independent claim 1, the Examiner contends on pages 3 of the Office

Action of 03/09/2007 that Thompson in column 1, lines 46-47 teaches Applicants' claimed

feature of **extracting *task* information from said operational *metadata***. However, col. 1, lines

46-47 merely teach the extraction of **financial "data from operational systems"** as part of a

"**consolidated process**" in an enterprise system. Thompson makes it clear that this data from

operational systems is financial data. See, for example, in the same column (i.e., column 1)

where Thompson further mentions using this "data from operational system" to provide "**an**

overall view of the condition and performance of the enterprise".

Applicants respectfully maintain that **extracting financial data from an operational**

system is NOT the same as **extracting *task* information from operational *metadata* obtained**

from a data-warehousing environment. Metadata, for example, is defined in Webopedia

(www.webopedia.com) as: "Data about data. Metadata describes how and when and by whom a

particular set of data was collected, and how the data is formatted. Metadata is essential for

understanding information stored in data warehouses and has become increasingly important in XML-based Web applications." Applicants maintain that Thompson fail to teach or suggest the feature of **extracting task information from operational metadata** or **extracting task information from changes in operational metadata**.

Next, the Examiner also contends on page 3 of the Office Action of 03/09/2007 that column 4, lines 31-32 of Thompson teaches Applicants' claimed feature of **extracting task information from said metadata**. Column 4, lines 31-32 of Thomson merely recites "**retrieving operational data *from the data source application* using a data flow plan**" (emphasis added). Such **retrieval of operational data from a *data source*** CANNOT be equated to **extracting task information from metadata**.

Further, the Examiner asserts that lines 4, lines 56-59 teach the feature of "**refreshing said buffer with changes in said operational metadata**". Further, the Examiner states on the same page of the Office Action of 03/09/2007 that the Abstract of Thompson teaches the feature of "**moving task information from said buffer to an archive**".

Column 4, lines 56-59 of Thompson is reproduced below:

> "Sometimes the loading of information comprises one of: a *round-robin approach* used for refresh processing and extracting information from permanent tables; and a *see-saw approach* used for non-refresh processing and extracting information from

temporary tables." (emphasis added).

It can be seen that the Examiner's citation **merely teaches the two ways anticipated by Thompson for loading information – one being the round-robin approach and the other being the see-saw approach**.

Similarly, the Abstract of Thompson merely teaches a transformation and staging server that obtains data from the data source application via requests and <u>places the data into temporary staging tables to prepare for the transformation and cleansing process prior to movement of the data to the data warehouse server</u>.

Applicants are unsure how the Examiner is interpreting the use of a round-robin approach and the placement of data in <u>temporary</u> staging tables to read on Claim 1's feature of "refreshing said buffer with ***changes* in said operational metadata** and **moving task information from said buffer to an archive**".

The secondary reference used by the Examiner (i.e., Norcott) merely teaches a method for **synchronous change data capture**, comprising the steps of: generating a transaction identifier that uniquely identifies a transaction, for each operation in a transaction, recording change data for the operation and the transaction identifier in a first database object, and during a commit of the transaction, recording the transaction identifier and a system change number in a second database object.

8

However, Norcott fails to remedy many of the shortcomings of the Thompson patents.

Hence, Applicants respectfully assert that an improper rejection was issued with regards to independent claim 1 as the art of record (i.e., Thompson and Norcott) fail to teach many of the features of claim 1.

The above-mentioned argument for claim 1 substantially applies for independent claim 30 as it recites an article of manufacture with similar features. Hence, Applicants respectfully assert that an improper rejection was issued with regards to independent claim 30 as the art of record (i.e., Thompson and Norcott) fail to teach many of the features of claim 1.

With respect to claim 10, the Examiner cites column 2, lines 2-4 and column 5, lines 2-4 of the Thompson patent as teaching the feature of "***uniquely* identifying *each task* within a run**". Column 2, lines 2-4 of Thompson merely teaches an "Enterprise Management System" that includes 1) <u>data extraction and movement</u>, 2) <u>data transformation and cleansing</u>, 3) <u>database updated and tuning</u>, and4) database access. Applicants are unsure how the Examiner is interpreting any of the above-mentioned four tasks as reading on the feature of **uniquely identifying tasks within a run**. Applicants maintain that none of the four tasks mentioned in column 2, lines 2-4 can be equated to Applicants claimed feature of identifying tasks within the run.

Similarly, the other citation by the Examiner of column 5, lines 2-4 (of Thompson) merely teaches **updating load statistics metadata for each table**. Applicants are once again unsure how such a step of uploading statistics can be equated to Applicants' claimed feature of uniquely identifying tasks within a run.

Also, with respect to claim 10, the Examiner equates Thompson column 5, lines 2-4 as teaching Applicants' feature of **selecting one or more of said uniquely identified tasks to monitor**. However, column 5, lines 2-4, by Thompson's own admission, merely teaches an **indication that information is in a loading state**. Such a feature of a state associated with loading/unloading CANNOT be equated to Applicants' claimed feature of **selecting or more uniquely identified tasks to *monitor***. Applicants maintain that the Thompson or Norcott fail to teach any monitoring of uniquely identified tasks, wherein changes to **operational metadata associated with a monitored task** is calculated and stored in a buffer, wherein **the contents of the buffer is moved into an archive**.

With respect to claim 10, the Examiner once again repeats the assertion that the Thompson abstract teaches "storing results of said calculating step in a buffer and moving selected buffer to an archive". The Examiner is once again reminded that the Abstract merely teaches the transformation and staging server that obtains data from the data source application via requests and **places the data into temporary staging tables to prepare for the transformation and cleansing process prior to movement of the data to the data warehouse server**.

Applicants are again unsure regarding how the Examiner is interpreting the placement of data in <u>temporary</u> staging tables to read on Claim 1's feature of "**storing results of said calculating step in a buffer and moving selected buffer to an archive**".

Hence, at least for the reasons state above, Applicants respectfully assert that an improper rejection was issued with regards to independent claim 10 as the art of record (i.e., Thompson and Norcott) fail to teach the claimed features of claim 10.

Further with respect to independent claim 29, the Examiner relies on the Norcott reference to asserts that the triggers 115 shown in Norcott can be equated to the "trigger mechanisms" of Applicants' claim 29. However, by Norcott's own admission, triggers 115 are merely means for "**firing an action routing**" when "**rows are inserted, updated, or deleted**". However, the trigger mechanism of claim 29 is specific in that the **trigger mechanism is** *attached to a staging table that stores task information extracted from operational metadata*. Such a trigger mechanism is neither taught nor suggested by the Thompson or Norcott references.

Hence, at least for the reasons state above, Applicants respectfully assert that an improper rejection was issued with regards to independent claim 29 as the art of record (i.e., Thompson and Norcott) fail to teach the claimed features of claim 29.

The above-mentioned arguments for claims 1, 10, 29, and 30 substantially apply to dependent claims 2-9, 11-17, 19-28, and 31-38 as they inherit all the features of the claim from which they depend.

Hence, at least for that reason, Applicants contend that an improper rejection was issued with regards to dependent claims 2-9, 11-17, 19-28, and 31-38.

Since the Examiner has failed to establish a prima facie case of obviousness under 35 U.S.C. §103(a) with respect to pending claims 1-17 and 19-38, Applicants maintain that an improper 35 U.S.C. §103(a) rejection was issued with regards to pending claims 1-17 and 19-38.

SUMMARY

As has been detailed above, none of the references, cited or applied, provide for the specific claimed details of applicant's presently claimed invention, nor render them obvious. It is believed that this case is in condition for allowance and reconsideration thereof and early issuance is respectfully requested.

As this Appeal Brief has been timely filed within the set period of response, no fee for extension of time is required. However, the Commissioner is hereby authorized to charge any deficiencies in the fees provided, including extension of time, to Deposit Account No. 09-0460.

Respectfully submitted by
Applicant's Representative,

/ramraj soundararajan/

Ramraj Soundararajan
Reg. No. 53,832

IP Authority, LLC.
9435 Lorton Market Street
#801
Lorton, VA 22079
(571)642-0033

August 13, 2007

13

Claims Appendix:

1. **(Previously Presented)** A method for archiving task information obtained from a data-warehousing environment comprising steps of:

 a. obtaining changes in operational metadata from said data-warehousing environment,

 b. extracting task information from said operational metadata,

 c. storing said extracted task information in a buffer,

 d. refreshing said buffer with changes in said operational metadata, and

 e. moving task information from said buffer to an archive,

 said archived task information used in data analysis and mining.

2. **(Original)** A method for archiving task information, as per claim 1, wherein said task is an extract, transform, load (ETL) task.

3. **(Original)** A method for archiving task information, as per claim 1, wherein said buffer is a staging table.

4. **(Original)** A method for archiving task information, as per claim 1, wherein said changes in operational metadata are obtained via a trigger mechanism.

14

5.   (Original) A method for archiving task information, as per claim 2, wherein said ETL task information comprises any of: ETL task execution statuses, run identification numbers, definitions, control flows, and execution schedules.

6.   (Original) A method for archiving task information, as per claim 2, wherein said archive is queried to report any of: completed tasks, pending tasks, duration of execution, error codes and messages, scheduling problems, scheduling changes, overdue ETL task run schedules, and overdue ETL task misses.

7.   (Previously Presented) A method for archiving task information, as per claim 2, wherein content of said archive is extracted from and stored in one or more tables.

8.   (Original) A method for archiving task information, as per claim 7, wherein said tables indicate any of: ETL task errors, completed tasks, task temporary status, and task scheduled.

9.   (Original) A method for archiving task information, as per claim 8, wherein said tables are queried to generate reports comprising any of: sequence of task executed in a process, last task executed, task or tasks failed, duration of execution of tasks in a process, task or tasks retried, and statistics associated with a task run or runs, errors associated with failed tasks, tasks failing with a specified error, task run schedule, de-scheduled tasks, and tasks having a specified temporary status.

10. (Previously Presented) A method for capturing and recording task information obtained from a data-warehousing environment for analysis, archival, and mining comprising steps of:

    a. uniquely identifying each task within a run,

    b. selecting one or more of said uniquely identified tasks to monitor,

    c. capturing data-warehousing population activities dynamically by

        i. obtaining operational metadata containing task information relevant to said selected task or tasks,

        ii. calculating changes in operational metadata,

        iii. storing results of said calculating step in a buffer, and

moving selected buffer data to an archive, said archive used in data analysis and mining.

11. (Original) A method for capturing and recording task information, as per claim 10, wherein said task is an extract, transform, load (ETL) task.

12. (Original) A method for capturing and recording task information, as per claim 10, wherein said buffer is a staging table.

13. (Original) A method for capturing and recording task information, as per claim 10, wherein either one of a system or a user performs said selecting step.

16

14. (Original) A method for capturing and recording task information, as per claim 10, wherein said operational metadata and changes in operational metadata are obtained via a trigger mechanism.

15. (Original) A method for capturing and recording task information, as per claim 14, wherein said trigger mechanism is attached to said operational metadata and to said buffer.

16. (Original) A method for capturing and recording task information, as per claim 14, wherein said trigger mechanism attached to operational metadata is activated by either changes to said selected task in said operational metadata or by termination of said selected task.

17. (Original) A method for capturing and recording task information, as per claim 15, whereupon termination of said selected task; said task status information is extracted from said operational metadata, if said selected task terminates with a failure or warning status, then error messages associated with said selected task or tasks are also extracted from said operational metadata, and said extracted task information is transformed into a format necessary for storage in said buffer.

18. (Cancelled)

19. (Original) A method for capturing and recording task information, as per claim 17, wherein upon termination of said selected task:

    a.   said trigger mechanism attached to said operational metadata is activated,

    b.   said buffer is refreshed with changes in said operational metadata before said trigger mechanism was activated,

    c.   said archive is emptied into a backup medium or media, and

said buffer data relevant to said selected task is moved from said buffer to said archive.

20. (Original) A method for capturing and recording task information, as per claim 19, wherein the granularity of data moved from said buffer to said archive is variable.

21. (Original) A method for capturing and recording task information, as per claim 19, wherein refresh operations on said buffer occur in response to the activation of said trigger mechanisms attached to said operational metadata.

22. (Original) A method for capturing and recording task information, as per claim 19, wherein said archive is queried to report any of: completed tasks, pending tasks, duration of execution, error codes and message, scheduling problems, scheduling changes, and overdue task runs, and overdue task misses.

23. (Original) A method for capturing and recording task information, as per claim 18, wherein said backup step comprises: selecting archive data to backup, backing up said selected archive

data, extracting said selected archive data from said archive, filtering said selected archive data from said archive, and moving to a table said filtered archive data.

24. (Original) A method for capturing and recording task information, as per claim 18, wherein said archive is backed up at configured intervals.

25. (Original) A method for capturing and recording task information, as per claim 19, wherein said buffer data to be backed up is associated with a current timestamp.

26. (Original) A method for capturing and recording task information, as per claim 25, wherein said current timestamp is utilized in backup restoration.

27. (Original) A method for capturing and recording task information, as per claim 23, wherein said tables indicate any of: tasks completed, task errors, task temporary statuses, and tasks scheduled.

28. (Original) A method for capturing and recording task information, as per claim 27, wherein said tables are queried to generate reports comprising any of: sequence of tasks executed in a process, last task executed, task or tasks failed, duration of execution of tasks in a process, task or tasks retried, and statistics associated with a task run or runs, errors associated with failed tasks, tasks failing with a specified error, task run schedule, de-scheduled tasks, and tasks having a specified temporary status.

19

29. (Previously Presented) A data-warehousing environment system for capturing and recording task information, said data-warehousing environment implemented in computer storage, said computer storage storing:

    a. task information extracted from operational metadata,

    b. trigger mechanisms attached to said operational metadata,

    c. staging table storing said task information,

    d. trigger mechanisms attached to said staging table, and

    e. an archive table storing task information from said staging table.

30. (Previously Presented) An article of manufacture comprising a computer storage medium having computer readable program code embodied therein which implements the archiving of task information obtained from a data-warehousing environment comprising modules to execute the steps of:

    a. obtaining changes in operational metadata from said data-warehousing environment,

    b. extracting task information from said operational metadata,

    c. storing said extracted task information in a buffer,

    d. refreshing said buffer with changes in said operational metadata, and

moving task information from said buffer to an archive.

31. (Original) An article of manufacture, as per claim 30, wherein said task is an extract, transform, load (ETL) task.

32. (Original) An article of manufacture, as per 30, wherein said buffer is a staging table.

33. (Original) An article of manufacture, as per claim 30, wherein said medium further comprises computer readable program code obtaining changes in operational metadata via a trigger mechanism.

34. (Original) An article of manufacture, as per claim 31, wherein said ETL task information comprises any of: ETL task execution statuses, run identification numbers, definitions, control flows, and execution schedules.

35. (Original) An article of manufacture, as per claim 31, wherein said medium further comprises computer readable program code querying said archive to report any of: completed tasks, pending tasks, duration of execution, error codes and messages, scheduling problems, scheduling changes, overdue ETL task run schedules, and overdue ETL task misses.

36. (Original) An article of manufacture, as per claim 35, wherein said medium further comprises computer readable program code extracting and storing content of said archive into one or more tables.

37. (Original) An article of manufacture, as per claim 36, wherein said tables indicate any of:

ETL task errors, completed tasks, task temporary status, and task scheduled.

38. (Original) An article of manufacture, as per claim 37, wherein said medium further comprises computer readable program code querying said tables to generate reports comprising any of: sequence of task executed in a process, last task executed, task or tasks failed, duration of execution of tasks in a process, task or tasks retried, and statistics associated with a task run or runs, errors associated with failed tasks, tasks failing with a specified error, task run schedule, de-scheduled tasks, and tasks having a specified temporary status.

Evidence Appendix

None

Related Proceedings Appendix

None